

Week 3

i=2

i>=5

False



False

we would get a false because two is less than five.

## Question

What is the output of the following code segment:

```
1  i=6  
2  i<5
```

- True
- False
- SyntaxError: can't assign to literal

✓ **Correct**  
correct

Skip

Continue

i=2

i!=6

True



True



## Question

What is the result of the following:  $5! = 5$

- False
- True

✓ **Correct**

correct, this is the inequality operator

Skip

Continue

`"AC/DC" == "Michael Jackson"`

**False**

`"AC/DC" == "Michael Jackson"`

False

`"AC/DC" != "Michael Jackson"`

True

## Question

What is the output of the following code segment: 'a'=='A'

- False
- True

✓ **Correct**

correct, the equality operator is case sensitive

Skip

Continue



```
age=17
```

```
if (age>18):
```

```
    print("you can enter" )
```

```
    print("move on")
```

False



In this case, it will just print "move on".

```
age=19
```

```
if (age>18):
```

```
    print("you can enter" )
```

```
    print("move on")
```

True

ACDC

you can enter

Therefore, the program will execute the statement to print "you will enter".

## Question

age

if (

in the video, if `age=18` what would be the result

- move on
- you can enter

✓ **Correct**

correct

pri

Skip

Continue

```
if (age>18):  
    print("you can enter" )  
  
else:  
    print("""go see Meat Loaf" )  
  
    print("move on")
```

```
if (age>18):
```

```
    print("you can enter" )
```

```
else:
```

```
    print("""go see Meat Loaf" )
```

```
    print("move on")
```

We then add the expression we would like to execute with an indent.



```
age=17
```

```
if (age>18):
```

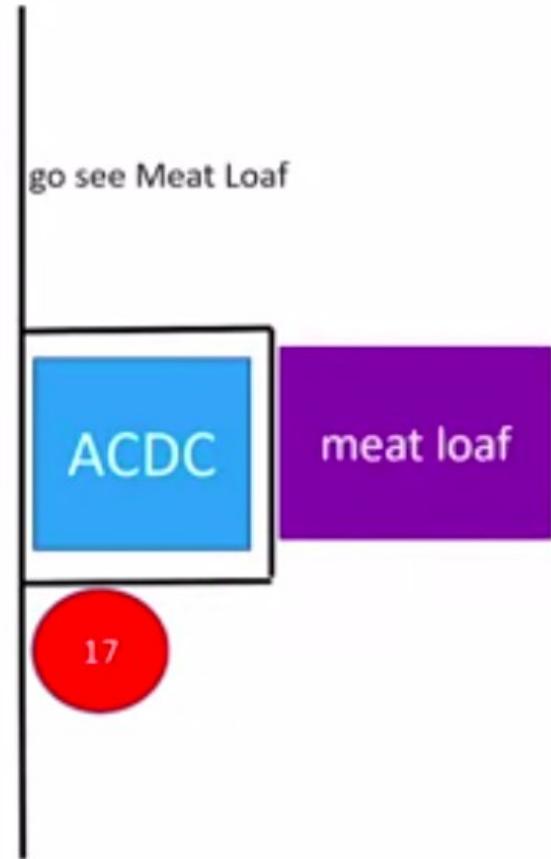
False

```
    print("you can enter" )
```

```
else:
```

```
    print("go see Meat Loaf" )
```

```
    print("move on")
```



```
age=19
```

```
if (age>18):
```

True

```
    print("you can enter" )
```

```
else:
```

```
    print("go see Meat Loaf" )
```

```
print("move on")
```

ACDC

meat loaf

```
age=19
```

```
if (age>18):
```

True

```
    print("you can enter" )
```

```
else:
```

```
    print("go see Meat Loaf" )
```

```
print("move on")
```

you can enter

ACDC

meat loaf

```
age=19
```

```
if (age>18):
```

True

```
    print("you can enter" )
```

you can enter

```
else:
```

```
    print("go see Meat Loaf" )
```

ACDC

meat loaf

```
print("move on")
```

19

## Question

age

if (

else

priv

the

in the video what would be the result if we set the variable age as follows: **age= -10**

- go see Meat Loaf

move on

✓ Correct

- you can enter

move on

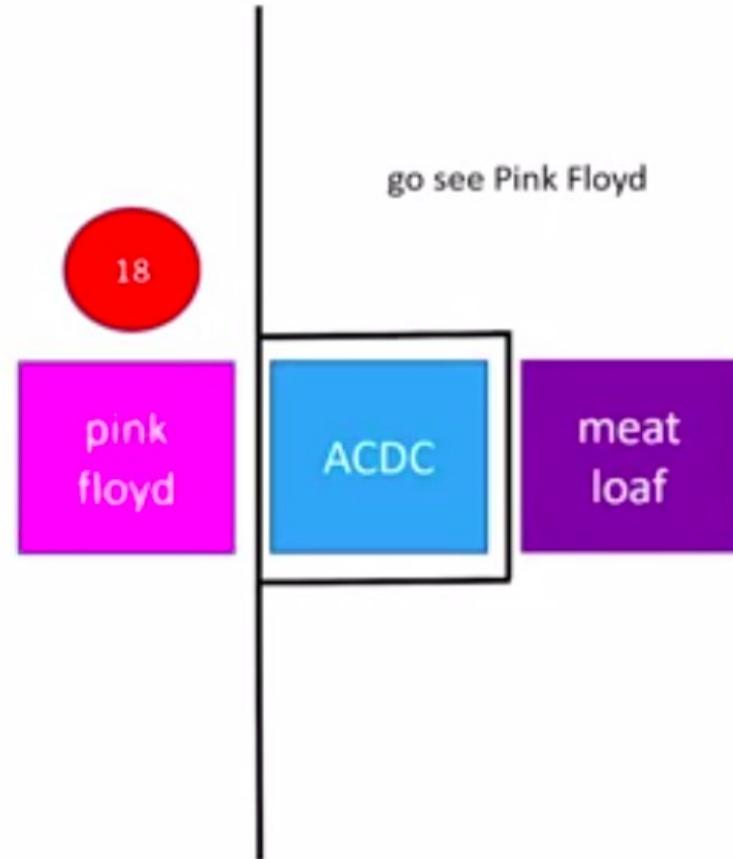
Skip

Continue



```
age=18  
if (age>18):  
    print("you can enter" )  
  
elif(age==18):  
    print("go see Pink Floyd" )  
  
else:  
    print("go see Meat Loaf" )  
  
print("move on")
```

True



So the condition of the elif statement is checked.

```
age=18  
if (age>18):  
    print("you can enter" )  
  
elif(age==18):  
    print("go see Pink Floyd" )  
  
else:  
    print("go see Meat Loaf" )  
  
print("move on")
```

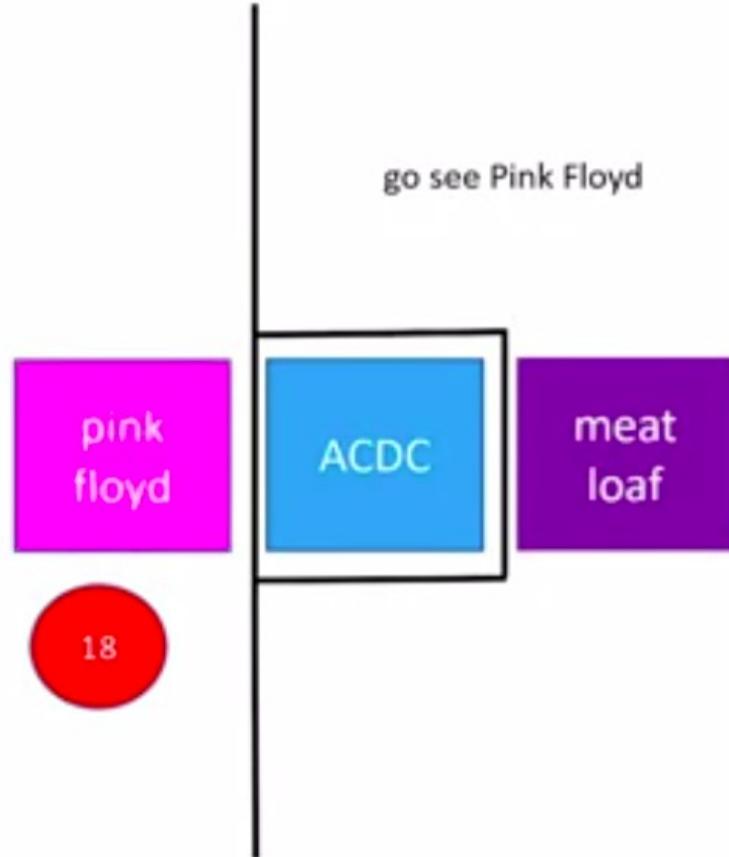
True



go see Pink Floyd

```
age=18  
if (age>18):  
    print("you can enter" )  
  
elif(age==18):  
    print("go see Pink Floyd" )  
  
else:  
    print("go see Meat Loaf" )  
  
print("move on")
```

True



# Logic Operators

not

False

**not(True)**

False

the result is a false.



# Logic Operators

not

True

**not(False)**

True

the result is a true.

# Logic Operators: OR

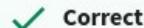
A	B	A or B
False	False	False
False	True	True
True	False	True
True	True	True

We can use this table to represent the different values.

## Question

what is the result of the following: **True or False**

- True, an **or** statement is only False if all the Boolean values are False
- False



correct,an **or** statement is only False if all the Boolean values are False

Skip

Continue

The follo

e 70s' or

```
album_year = 1990
```

```
If (album_year < 1980) or (album_year > 1989):
```

```
    print ("The Album was made in the 70 's or 90's")
```

```
else:
```

```
    print("The Album was made in the 1980's ")
```

The Album was made in the 70's or 90's

album\_year < 1980

False

1990

or

True

1990

album\_year > 1989

True

True

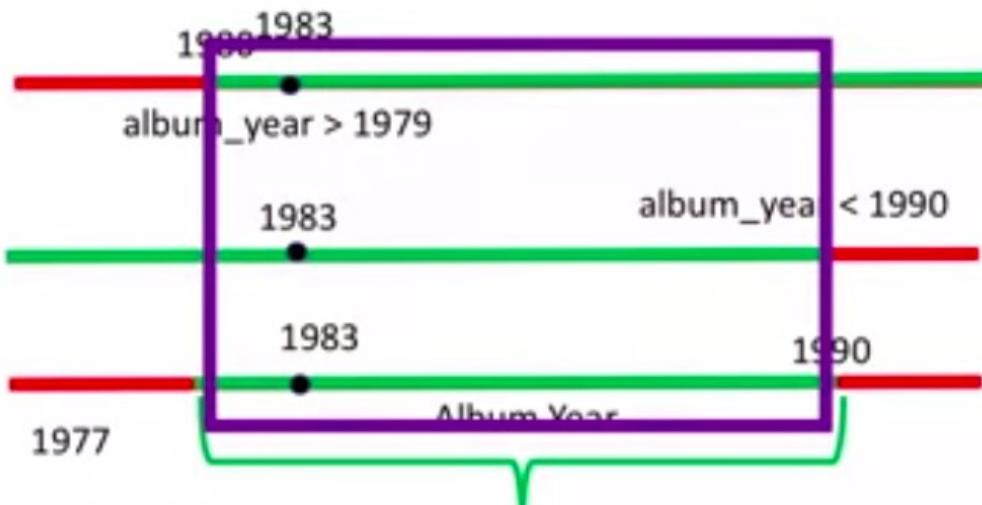
# Logic Operators: AND

A	B	A & B (AND)
False	False	False
False	True	False
True	False	False
True	True	True

```
album_year = 1983
```

```
if(album_year > 1979) and (album_year < 1990):
```

```
    print ("This album was made in the 80's ")
```



[Back](#)

## Practice Quiz

Practice Quiz • 21 min • 7 total points

1. what is the result of the following: **1=2**

1 / 1 point

- SyntaxError:can't assign to literal
- True
- False

 **Correct**

correct, this statement is a syntax error

2. What is the output of the following code segment:

1 / 1 point

i=6

i&lt;5

[Back](#)

## Practice Quiz

Practice Quiz • 21 min • 7 total points

3. What is the result of the following: **5!=5**

1 / 1 point

- False  
 True

 **Correct**

correct, this is the inequality operator

4. What is the output of the following code segment: 'a'=='A'

1 / 1 point

- False  
 True

 **Correct**

correct, the equality operator is case sensitive

[Back](#)

## Practice Quiz

Practice Quiz • 21 min • 7 total points

5. in the video, if **age=18** what would be the result

1 / 1 point

- move on  
 you can enter

 **Correct**  
correct

6. in the video what would be the result if we set the variable age as follows: **age= -10**

1 / 1 point

- go see Meat Loaf  
move on  
 you can enter  
move on

 **Correct**

[Back](#)

## Practice Quiz

Practice Quiz • 21 min • 7 total points

6. in the video what would be the result if we set the variable age as follows: **age= -10**

1 / 1 point

 go see Meat Loaf

move on

 you can enter

move on

 **Correct**

7. what is the result of the following: **True or False**

1 / 1 point

 True, an **or** statement is only False if all the Boolean values are False False **Correct**correct,an **or** statement is only False if all the Boolean values are False

*range(N)*



[0, ..., N - 1]

If the input is a positive integer,

*range(N)*

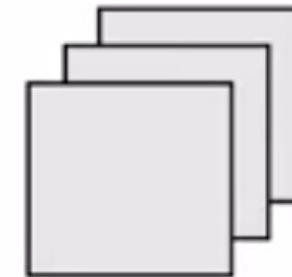


[0, ..., N - 1]

*range*(10,15)

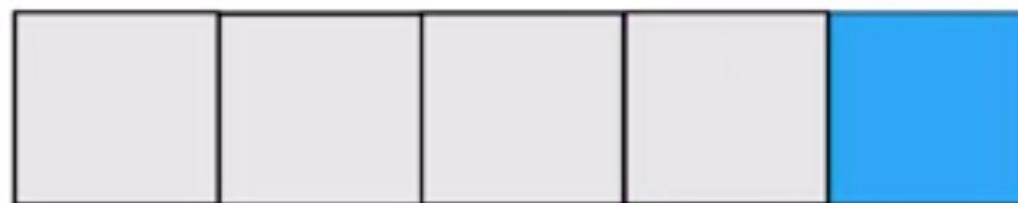


[10, 11, 12, 13, 14]



For square 0 in squares, square 0 = white square  
For square 1 in squares, square 1 = white square





squares	0	1	2	3	4
---------	---	---	---	---	---

For square 0 in squares, square 0 = white square

For square 1 in squares, square 1 = white square

For square 2 in squares, square 2 = white square

For square 3 in squares, square 3 = white square

For square 4 in squares, square 4 = white square

The only thing that changes is the index of the square we are referring to.



squares	0	1	2	3	4
---------	---	---	---	---	---

For square 0 in squares, square 0 = white square

For square 1 in squares, square 1 = white square

For square 2 in squares, square 2 = white square

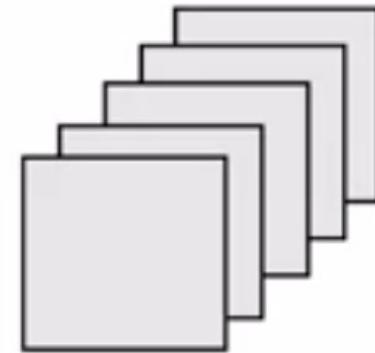
For square 3 in squares, square 3 = white square

For square 4 in squares, square 4 = white square

The only thing that changes is the index of the square we are referring to.



squares	red	yellow	green	purple	blue
---------	-----	--------	-------	--------	------



We want to change the name of the color in each element to white.



squares	red	yellow	green	purple	blue
	0	1	2	3	4



```
squares=[“red”, “yellow ”, “green”, “purple”, “blue ”]
```

```
for i in range(0,5):
```

```
    squares[i]=“white”
```

## Question

what will be the output of the following:

```
1 for x in range(0,3):  
2     print(x)
```

0

1

2

0

1

2

3

✓ Correct

Skip

Continue



```
squares=[“red”, “yellow”, “green”]
```

```
for square in squares:
```

square



square



yellow

the value of square is yellow.

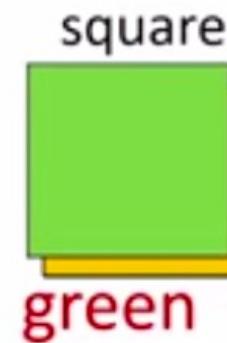




```
squares=["red", "yellow ", "green"]
```

```
for square in squares:
```

square



We then start the third iteration. For the final iteration, the value of square is  
green,



## Question

what is the output of the following:

```
1  for x in ['A', 'B', 'C']:  
2      |  
3      |  print(x+'A')
```



AA

BA

CA



A

B

C

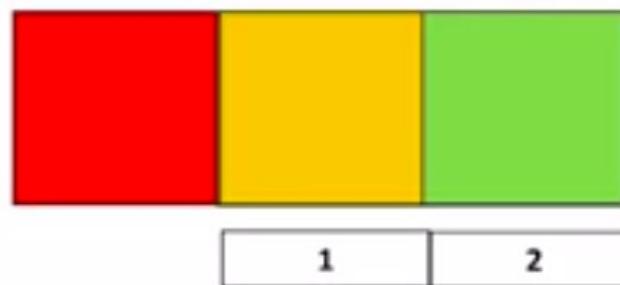


Correct

correct

Skip

Continue



```
squares=[“red”, “yellow ”, “green”]
```

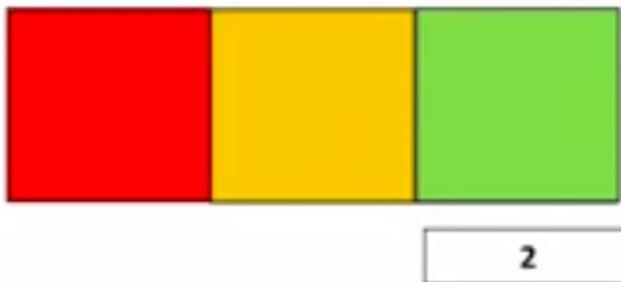
```
for i,square in enumerate(squares):
```

square  
i



square  
red

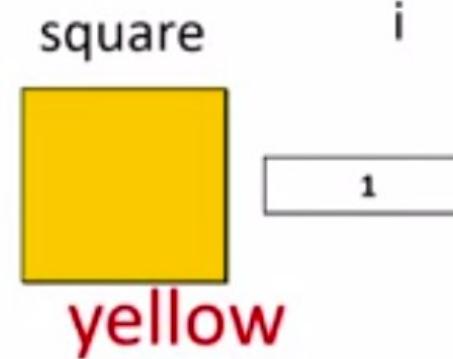
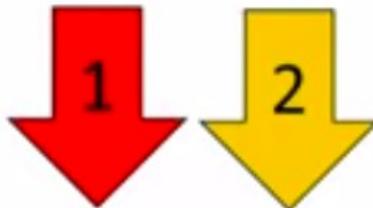
i  
0



```
squares=[“red”, “yellow”, “green”]
```

```
for i,square in enumerate(squares):
```

square  
i



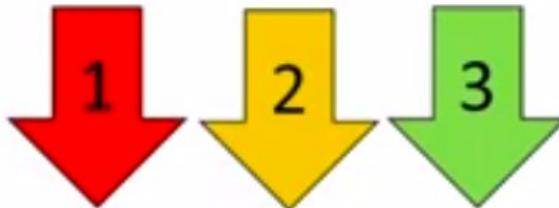
The value of the variable square is yellow, and



```
squares=[“red”, “yellow ”, ‘green’]
```

```
for i,square in enumerate(squares):
```

square  
i



square  
green

i  
2

## Question

what is the output of the following:

```
1 for i,x in enumerate(['A','B','C']):  
2 |   print(i,x)
```

0 A

1 B

2 C

AA

BB

CC

 **Correct**

correct

Skip

Continue

# while loops

While loops are similar to for loops but instead of executing

While

square



Copy

False

New  
squares



```
squares=['orange','orange','purple','orange','blue']
```

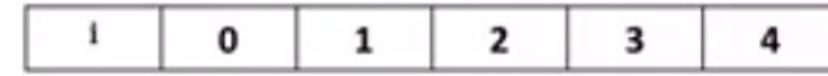
```
Newsquares=[]
```

```
i=0
```

```
while(squares[i]=='orange'):
```

```
    Newsquares.append(squares[i])
```

We append the value of the second element of the list squares to the list new squares.



```
while( squares[i] =='orange'):
```



```
squares=['orange','orange','purple','orange','blue']
```

```
Newsquares=[]
```

```
i=0
```

```
while(squares[i]=='orange'):
```

```
    Newsquares.append(squares[i])
```

```
i=i+1
```



```
while( squares[i] =='orange'):
```



```
squares=['orange','orange','purple','orange','blue']
```

```
Newsquares=[]
```

```
i=0
```

```
while(squares[i]=='orange'):
```

```
    Newsquares.append(squares[i])
```

```
    i=i+1
```



```
while( so  [i] =='orange'):
```



 [Back](#)

Practice Quiz • 9 min • 3 total points

1. what will be the output of the following:

1 / 1 point

```
for x in range(0,3):
```

```
print(x)
```

6

1

2

Q 0

1

2

3

 **Correct**

correct

[Back](#)

## Practice Quiz

Practice Quiz • 9 min • 3 total points

2. what is the output of the following:

1/1 point

```
for x in ['A','B','C']:  
    print(x+A)
```

 AA

BA

CA

 A

B

C

 **Correct**

correct

[Back](#)

## Practice Quiz

Practice Quiz • 9 min • 3 total points

3. what is the output of the following:

1 / 1 point

```
for i,x in enumerate(['A','B','C']):  
    print(i,x)
```

 0 A 1 B 2 C AA BB CC **Correct**

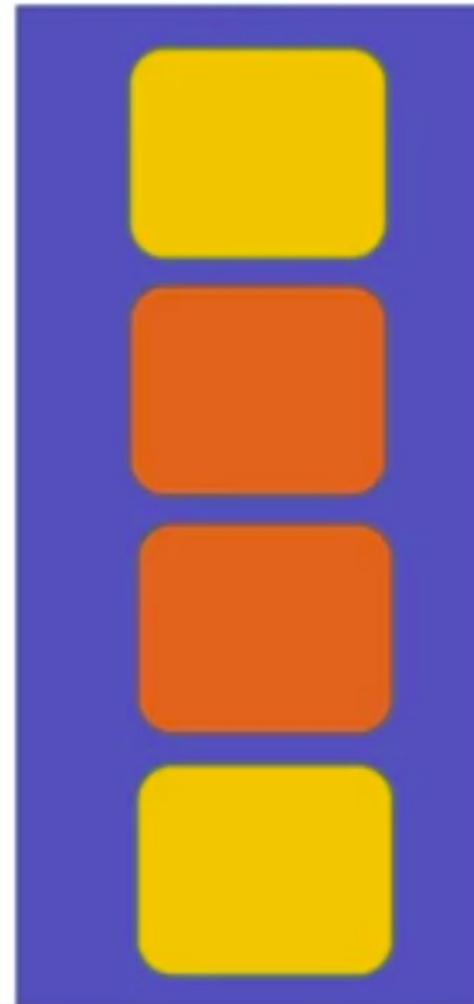
correct

```
def f1(input):
    """add 1 to input"""
    output=input+1;

    return output
```

```
def f2(input):
    """add 2 to input"""
    output=input+2;

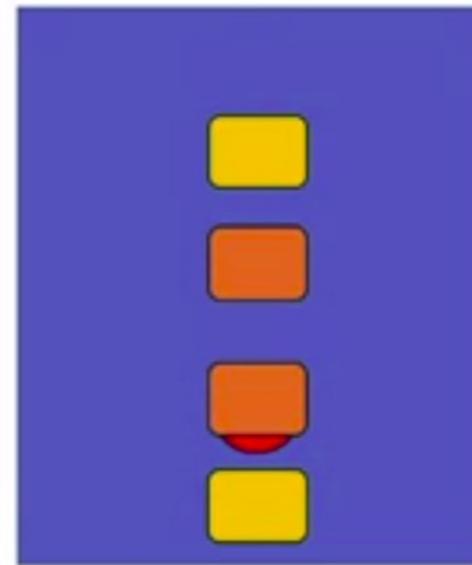
    return output
```



similar blocks of code. We can run the code using some input and get

```
def f1(input):
    """add 1 to input"""
    output=input+1;
    return output
```

```
def f2(input):
    """add 2 to input"""
    output=input+2;
    return output
```



# Len

```
album_ratings = [10.0,8.5,9.5,7.0,7.0,9.5,9.0,9.5]
```

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

```
L=len(album_ratings)
```

L:8

len

8

list, in this case, 8. The function sum takes in an iterable like

## Question

what does the following function return: `len(['A','B',1])`

- 3
- 2
- 4



**Correct**

correct, the function returns the number of elements in the list, in this case 3

Skip

Continue

## Sum

```
album_ratings = [10.0,8.5,9.5,7.0,7.0,9.5,9.0,9.5]
```

$$10.0 + 8.5 + 9.5 + 7.0 + 7.0 + 9.5 + 9.0 + 9.5$$

```
S = sum(album_ratings)
```

S:70

sum

70



## Question

what does the following function return: `len([sum([0,0,1])])`

- 1
- 0
- 3



correct

Skip

Continue

## Sorted vs Sort

```
album_ratings = [10.0,8.5,9.5,7.0,7.0,9.5,9.0,9.5]
```

```
sorted_album_rating = sorted(album_ratings)
```

```
sorted_album_rating:
```

```
[7.0, 7.0, 8.5, 9.0, 9.5, 9.5, 9.5, 10.0]
```

sorted

```
[7.0, 7.0, 8.5, 9.0, 9.5, 9.5, 9.5, 10.0]
```

The result is a new sorted list. If we look at the list album ratings, nothing

## Sorted vs Sort

```
album_ratings = [10.0,8.5,9.5,7.0,7.0,9.5,9.0,9.5]
```

```
sorted_album_rating = sorted(album_ratings)
```

```
sorted_album_rating:
```

```
[7.0, 7.0, 8.5, 9.0, 9.5, 9.5, 9.5, 10.0]
```

sorted

```
album_ratings :
```

```
[10.0,8.5,9.5,7.0,7.0,9.5,9.0,9.5]
```

```
[7.0, 7.0, 8.5, 9.0, 9.5, 9.5, 9.5, 10.0]
```

The result is a new sorted list. If we look at the list album ratings, nothing

## Question

What is the value of list L after the following code segment is run :

```
1 L=[1, 3, 2]  
2 sorted(L)
```

- L:[1,3,2]
- L:[1,2,3]
- L:[0,0,0]



**Correct**

correct, **sorted** is a function and returns a new list, it does not change the list **L**

Skip

Continue



## Sorted vs Sort

```
album_ratings = [10.0, 8.5, 9.5, 7.0, 7.0, 9.5, 9.0, 9.5]
```

```
album_ratings.sort()
```

```
album_rating:
```

```
[7.0, 7.0, 8.5, 9.0, 9.5, 9.5, 9.5, 10.0]
```

album\_ratings

Now that we have gone over how to use functions in Python, let's see how to  
build our own

```
def add1(a):
```

```
    b=a+1
```

```
    return b
```

```
add1(5)
```

```
6
```

```
c=add1(10)
```

```
c:11
```

## Question

def

from the video what is the value of c after the following:

- 1 c=add1(2)
- 2 c=add1(10)

,

- 3
- 11
- 14



**Correct**

correct, when you call the function the second time the value of c is reassigned.

It's custom

IBM Developer

Skip

Continue

anyone  
SKILLS NETWORK



```
def add1(a):
    """
    add 1 to a
    """
    b=a+1;
    return b
```

Documentation  
String

```
help(add1)
```

```
Help on function add1 in module __main__: add1(a) add 1 to a
```

to display the documentation as follows. This will printout the function name and the



# Multiple Parameters

- A function can have multiple parameters

```
def Mult(a,b):  
    c=a*b  
    return c
```

Mult(2,3)

6

Mult(10,3.14)

31.4



```
def Mult(a,b):  
    c=a*b  
    return c
```



2\*"Michael Jackson "

"Michael Jackson Michael Jackson "

```
Mult(2,"Michael Jackson ")
```

"Michael Jackson Michael Jackson "

Jackson is repeated two times. This is because the multiplication symbol

```
def NoWork():
    pass
print(NoWork())
```

None

Python doesn't allow a function to have an empty body, so we can use the keyword pass,

```
def add1(a):  
  
    b=a+1;  
  
    print(a, "plus 1 equals ",b)
```

return b

add1(2)

2 plus 1 equals 3

3

a	2
b	3
output of print(...)	2 plus 1 equals 3
value returned	3

Finally, the function returns the value of b, in this case, 3.

```
def printStuff(Stuff):  
    for i,s in enumerate(Stuff):  
        print("Album", i , "Rating is ", s)
```

```
album_ratings = [10.0,8.5,9.5]  
printStuff(album_ratings)
```

Album 0 Rating is 10

Album 1 Rating is 8.5

Stuff: [10.0, 8.5, 9.5]

Index: 

0	1	2
---	---	---

```
def printStuff(Stuff):  
    for i,s in enumerate(Stuff):  
        print("Album", i , "Rating is ", s)
```

```
album_ratings = [10.0,8.5,9.5]  
printStuff(album_ratings)
```

```
Album 0 Rating is 10  
Album 1 Rating is 8.5  
Album 2 Rating is 9.5
```

Stuff: [10.0, 8.5, 9.5]  
Index: 0 1 2



## Question

what is the output of the following lines of code:

```
1 def Print(A):
2     for a in A:
3         print(a+'1')
4
5 Print(['a','b','c'])
```

- a
- b
- c
- a1
- b1
- c1
- a1

Skip

Continue



# Scope

Global Scope

```
def AddDC(x):  
    x=x+"DC"  
    print(x)  
    return(x)
```

```
x="AC"  
z= AddDC(x)
```



# Scope: Local Variables

Global Scope

```
def Thriller():
    Date=1982
    return (Date)
```

Date=2017

```
print(Thriller())
```

```
print(Date)
```

The global variable date is set to 2017. When we call the function, we create a new



# Scope: Local Variables

Global Scope

```
def Thriller():
    Date=1982
    return (Date)
```

Date=2017

```
print(Thriller())
```

1982

```
print(Date)
```

Scope Thriller

Date  
1982

Date  
2017

The global value of the variable is 2017. Therefore, the value is set to 2017.

# Scope: Local Variables

Global Scope

```
def PinkFloyd():
    global ClaimedSales
    ClaimedSales ='45 million'
    return ClaimedSales
```

```
PinkFloyd()
```

```
print(ClaimedSales)
```

Claimed Sales  
'45 millions'

We call the function Pink Floyd. The variable claimed sales is set to the string

[Back](#)

## Practice Quiz

Practice Quiz • 15 min • 5 total points

1. What does the following function return: `len(['A','B',1])` ?

1 / 1 point

- 3
- 2
- 4

 Correct

Correct, the function returns the number of elements in the list, in this case 3.

2. What does the following function return: `len([sum([0,0,1])])` ?

1 / 1 point

- 1
- 0
- 3

 Correct

Correct

[Back](#)

## Practice Quiz

Practice Quiz • 15 min • 5 total points

3



Correct

Correct

3. What is the value of list L after the following code segment is run :

1 / 1 point

L=[1,3,2]

sorted(L)

 L:[1,3,2] L:[1,2,3] L:[0,0,0]

Correct

Correct, **sorted** is a function and returns a new list, it does not change the list **L**

[Back](#)

## Practice Quiz

Practice Quiz • 15 min • 5 total points

Correct, **sorted** is a function and returns a new list, it does not change the list **L**

4. From the video what is the value of c after the following:

1 / 1 point

`c=add1(2)``c=add1(10)` 3 11 14

Correct, when you call the function the second time the value of c is reassigned.

5. what is the output of the following lines of code:

1 / 1 point

[Back](#) Practice Quiz

Practice Quiz • 15 min • 5 total points

5. what is the output of the following lines of code:

1 / 1 point

```
def Print(A):
```

```
    for a in A:
```

```
        print(a+'1')
```

```
Print(['a','b','c'])
```

 a b c a1 b1 c1 a1

**Correct**

correct, the function concatenates a string

# Objectives

---

After watching this video, you will be able to:

- Explain Exception Handling
- Demonstrate the use of exception handling
- Understand the basics of exception handling

# Introduction

---

Please enter your  
name:

Please only enter  
letters

# Try...Except Statement

---

```
try:  
    myfile = open("myfile", "r")  
    myfile.write("File for exception  
handling.")  
except IOError:
```

# Try...Except Statements

```
try:  
    myfile = open("myfile", "r")  
    myfile.write("My file for  
exception handling.")  
except IOError:  
    print("Unable to open or read the  
data in the file.")
```

# Try...Except Statements

---

```
try:  
    myfile = open("myfile", "r")  
    myfile.write("My file for  
exception handling.")  
except IOError:  
    print("Unable to open or read the  
data in the file.")  
except:  
    print("Some other error  
occurred!")
```

# Try...Except Statements

```
try:  
    myfile = open("myfile", "r")  
    myfile.write("My file for exception  
handling.")  
except IOError:  
    print("Unable to open or read the data in  
the file.")  
except:  
    print("Some other error occurred!")
```

# Try...Except Statements

---

Some other  
error occurred!

# Try...Except...Else...Finally Statement

---

```
try:  
    getFile = open("myfile", "r")  
    getFile.write("My file for exception  
handling.")  
except IOError:  
    print("Unable to open or read the data in  
the file.")  
else:  
    print("The file was written successfully")
```

# Try...Except...Else...Finally Statement

---

```
try:  
    getFile = open("myfile", "r")  
    getFile.write("My file for exception  
handling.")  
except IOError:  
    print("Unable to open or read the data in  
the file.")  
else:  
    print("The file was written successfully")  
finally:  
    getFile.close()  
    print("File is now closed.")
```

Back

## Practice Quiz

Practice Quiz • 6 min • 2 total points

## Congratulations! You passed!

Grade received 100% To pass 50% or higher

[Go to next item](#)

## 1. Why do we use exception handlers?

1 / 1 point

- Terminate a program
- Catch errors within a program
- Write a file
- Read a file

Correct

## 2. What is the purpose of a try...except statement?

1 / 1 point

- Only executes if one condition is true

[Back](#)

## Practice Quiz

Practice Quiz • 6 min • 2 total points

- Terminate a program
- Catch errors within a program
- Write a file
- Read a file

 **Correct**

2. What is the purpose of a try...except statement?

1 / 1 point

- Only executes if one condition is true
- Executes the code block only if a certain condition exists
- Catch and handle exceptions when an error occurs
- Crash a program when errors occur

 **Correct**

Back

## Practice Quiz

Practice Quiz • 6 min • 2 total points

## 1. Why do we use exception handlers?

1 / 1 point

- Terminate a program
- Catch errors within a program
- Write a file
- Read a file

Correct

## 2. What is the purpose of a try...except statement?

1 / 1 point

- Only executes if one condition is true
- Executes the code block only if a certain condition exists
- Catch and handle exceptions when an error occurs
- Crash a program when errors occur

Correct

# Built-in Types in Python

- Python has lots of data types
- Types:
  - int: 1, 2, 567...
  - float: 1.2, 0.62...
  - String: 'a', 'abc', 'The cat is yellow'
  - List: [1, 2, 'abc']
  - Dictionary: {"dog": 1, "Cat": 2}
  - Bool: `False`, `True`
- Each is an **Object**

# Built-in Types in Python

---

- every **object** has:
  - a **type**
  - an internal data representation (a blueprint)
  - a set of procedures for interacting with the object (**methods**)
- an **object** is an **instance** of a particular **type**

Type 1

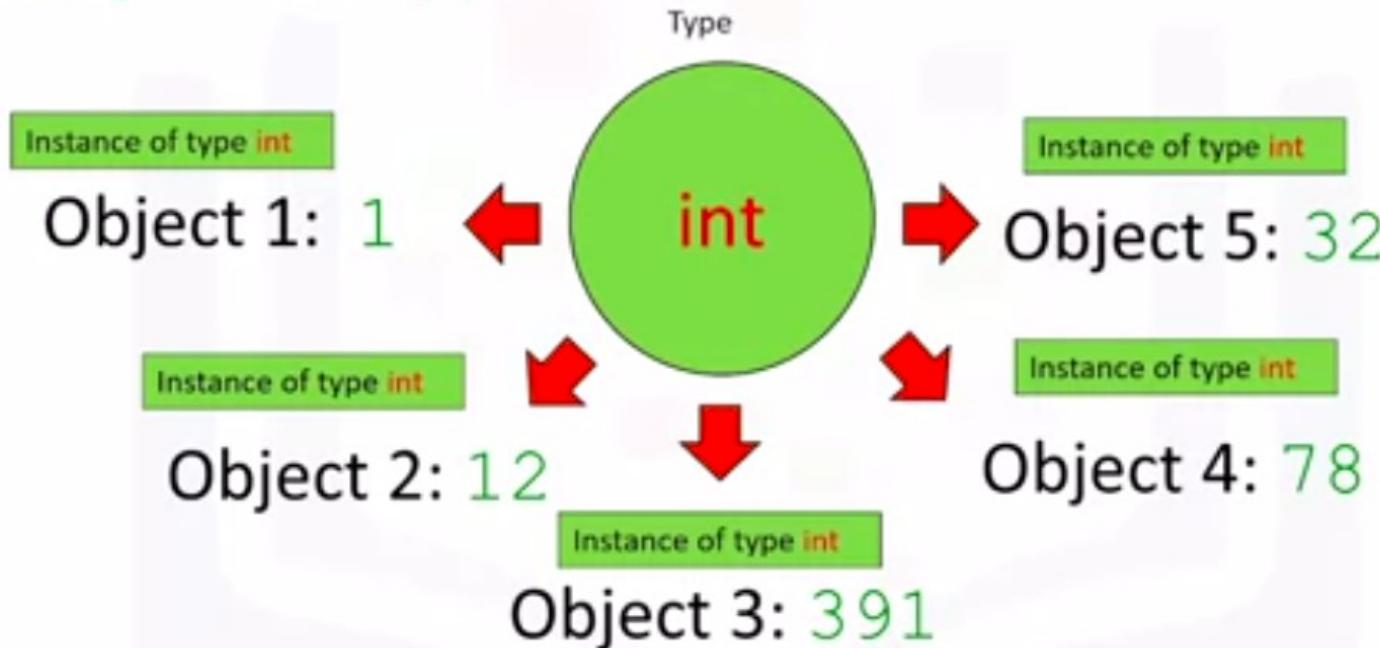
Object 1  
Object 2  
Object 3  
Object 4

Type 2

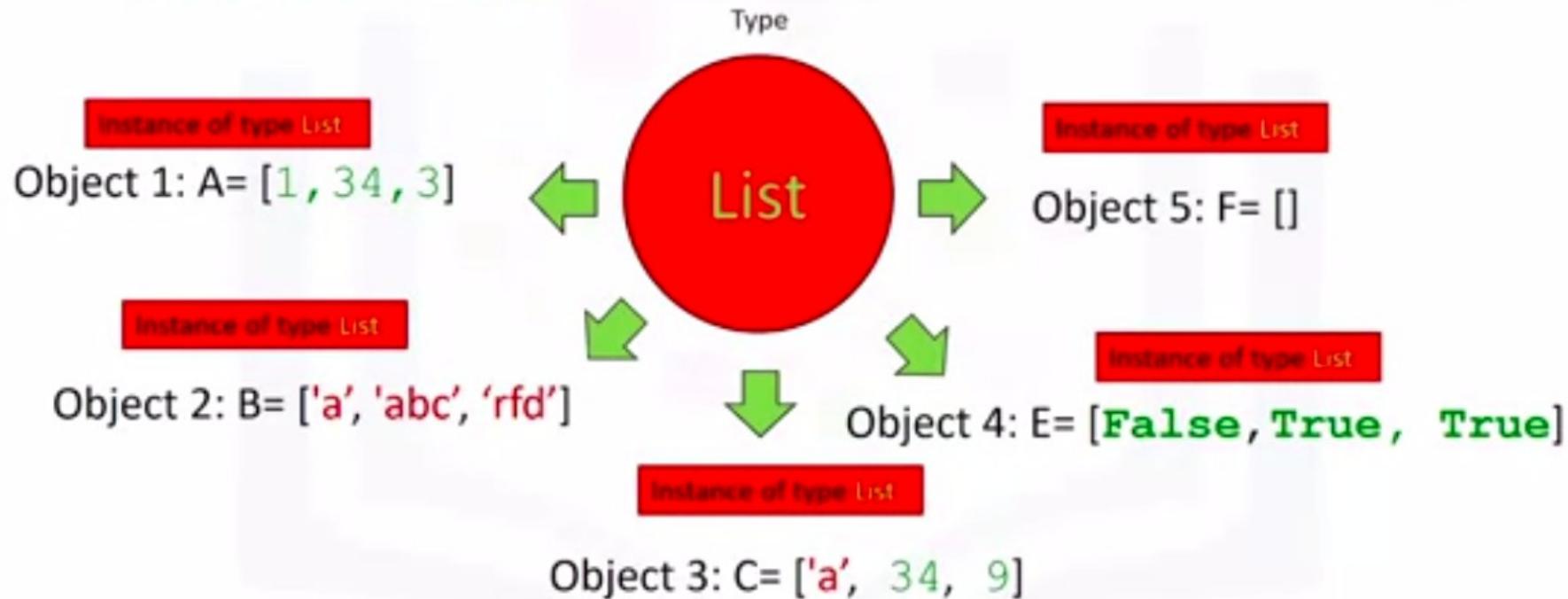
Object 1  
Object 2  
Object 3  
Object 4



# Objects: Type



# Objects: Type



# Objects: Type

- You can find the type of a object by using the command `type()`

```
>>type([1, 34, 3])  
<class 'list'>
```

Instance of type List



```
>>type(1)  
<class 'int'>
```

Instance of type int



```
>>type('The cat is yellow')  
<class 'str'>
```

Instance of type str



```
>>type( {"dog": 1, "Cat": 2})  
<class 'dict'>
```

Instance of type List



## Question

what is the type of the following:

1 `["a"]`

- str
- list

✓ **Correct**

correct, the "a" is surrounded by brackets so it is a list

Skip

Continue

## Methods

---

- A class or type's methods are functions that every instance of that class or type provides
- It's how you interact with the data in a object
- Sorting is an example of a method that interacts with the data in the object

Ratings=[10, 9, 6, 5, 10, 8, 9, 6, 2]

Ratings.sort()



Ratings=

[2, 5, 6, 6, 8, 9, 9, 10, 10]

[2, 5, 6, 6, 8, 9, 9, 10, 10]

.reverse()  
Method



Ratings=

[10, 10, 9, 9, 8, 6, 6, 5, 2]

## Question

what does a method do to an object

- changes or interacts with the object
- returns a new values



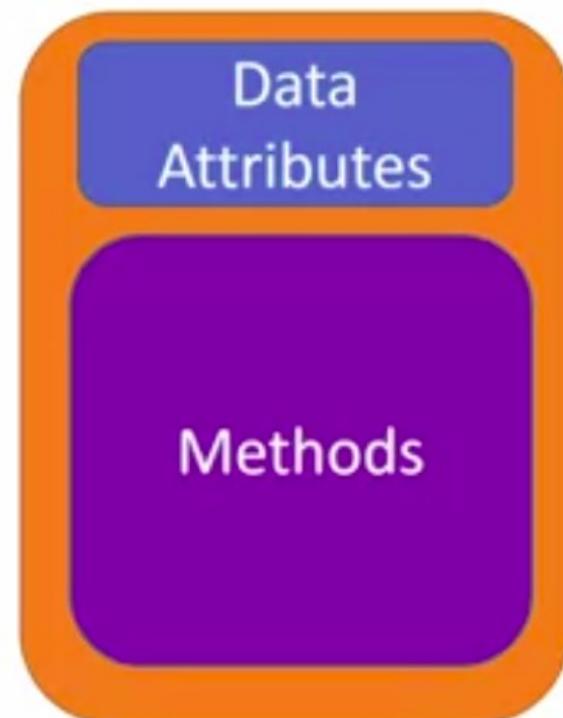
correct

Skip

Continue

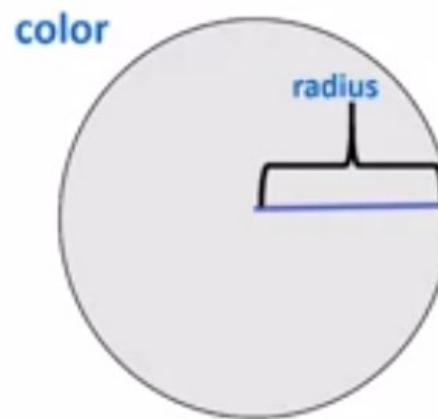
# Creating Your Own Types: Defining Classes

Class



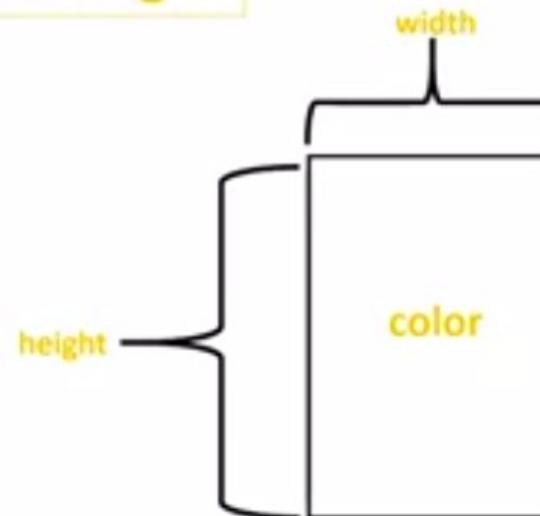
Objects or Instances of that Class

### Class Circle



Data Attributes: **radius, color**

### Class Rectangle



Data Attributes: **color, height and width**

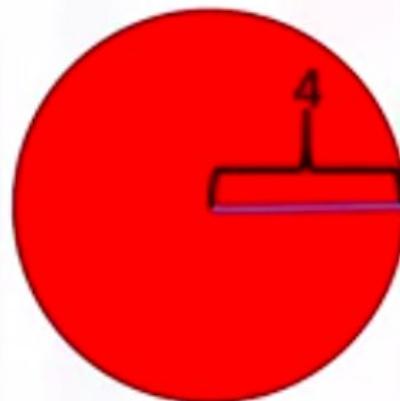
## Create a class: Circle

Name of Class  
class Circle (object):  
Class Definition      Class parent

## Create a class: Circle

Name of Class  
class Rectangle(object):  
Class Definition      Class parent

# Attributes and Objects



Object 1: Instance of type Circle

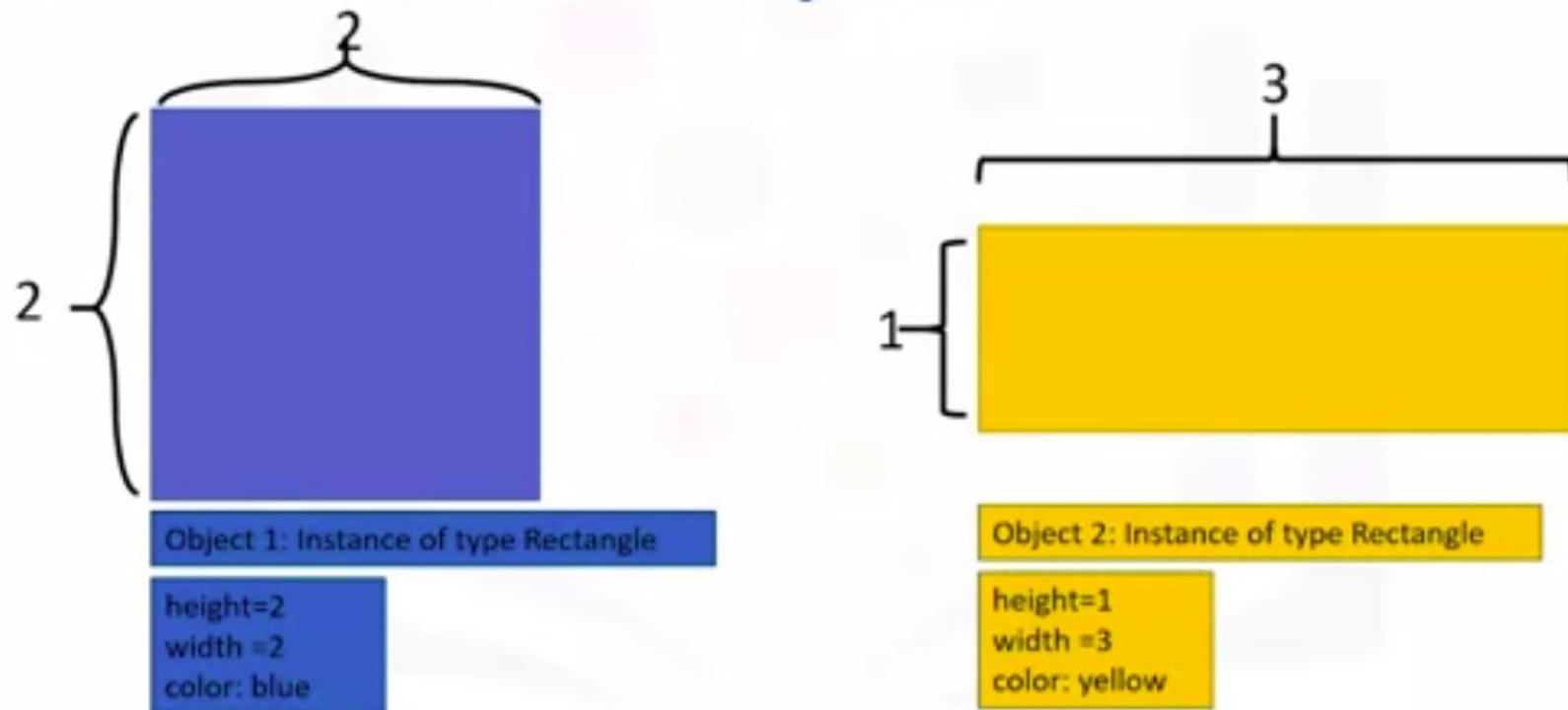
Data Attributes:  
radius=4  
color=red



Object 2: Instance of type Circle

Data Attributes:  
radius=2  
color=green

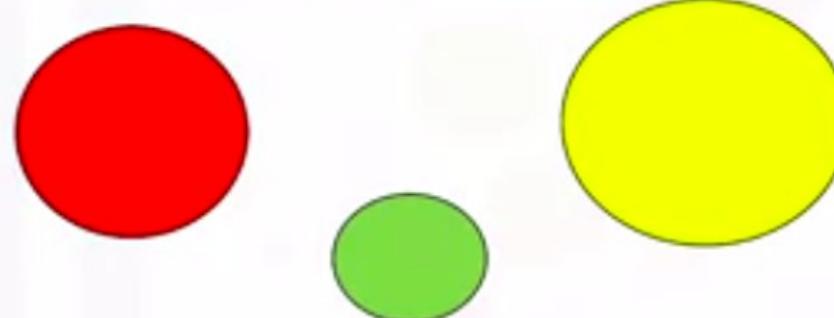
# Attributes and Objects



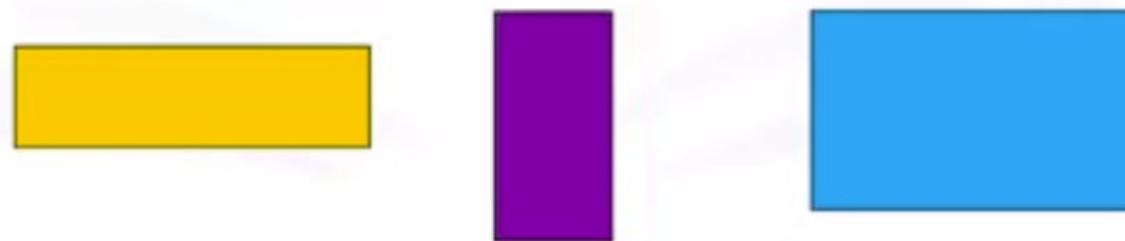
## Instances of a Class: objects

---

- We now have different objects of class circle or type circle



- We also have different objects of class rectangle or type rectangle



## Create a class: Circle

```
class Circle(object):
```

} Define your class

```
def __init__(self, radius , color):
```

```
    self.radius = radius;
```

```
    self.color = color;
```

} Data attributes used to  
Initialize each instance of  
the class

## Create a class: Circle

---

special method or constructor used to initialize data attributes

```
def __init__(self, radius , color):
```

```
    self.radius = radius;  
    self.color = color;
```



## Create a class: Circle

special method or constructor used to initialize data attributes

```
def __init__(self, radius , color):  
    self.radius = radius;  
    self.color = color;
```

The diagram illustrates the annotations for the `__init__` method:

- A bracket labeled "parameters" spans the entire list of parameters: `(self, radius , color)`.
- A bracket labeled "The self parameter" spans the first parameter: `self`.
- The assignment statements `self.radius = radius;` and `self.color = color;` are highlighted with a red rounded rectangle.

## Create a class: Rectangle

```
class Rectangle(object):
```

} Define your class

```
def __init__(self, color, height , width):
```

```
    self.height = height;
```

```
    self.width = width
```

```
    self.color = color;
```

} Initialize the object's  
Data attributes

## Create an Instance of a Class: Circle

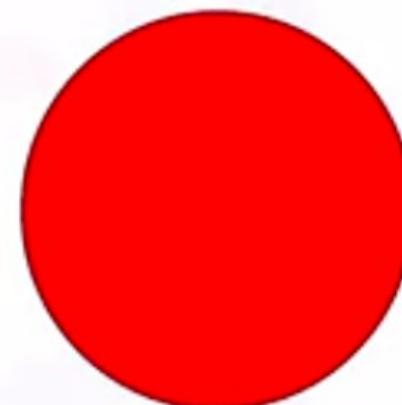
How to create an object of class circle:

Name of Class  
RedCircle =Circle (10, "red")  
Attributes

## Create an Instance of a Class: Circle

```
C1=Circle(10,'red')
```

```
class Circle(object):  
    def __init__(self, 10, 'red'):  
        self.radius = 10;  
        self.color = 'red';
```



```
self.radius = 10;  
self.color = 'red';
```

## Create an Instance of a Class: Circle

```
C1=Circle (10, "red")
```

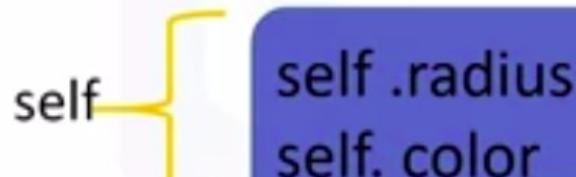
C1.radius

10

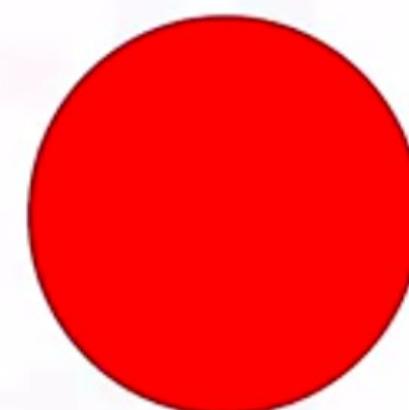
C1.color

"red"

self



self .radius  
self. color



```
self.radius = 10;  
self.color = 'red';
```

## Question

we create the object

```
1 Circle(3, 'blue')
```

what is the color attribute set to

2

'blue'

✓ **Correct**

correct

Skip

Continue

## Question

what is the radius attribute after the following code block is run:

```
1 RedCircle=Circle(10, 'red')
2 RedCircle.radius=1
```

- 10
- 1
- 'red'



correct, the line of code **RedCircle.radius=1** will change  
the radius

Skip

Continue

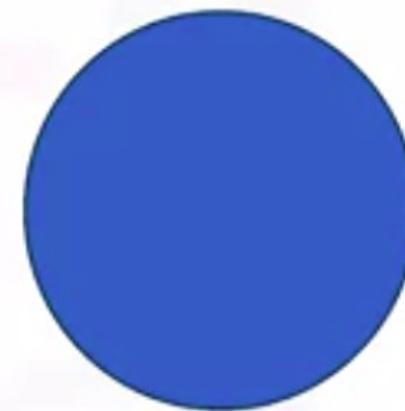
## Create an Instance of a Class: Circle

```
C1=Circle (10, "red")
```

```
C1.color="blue"
```

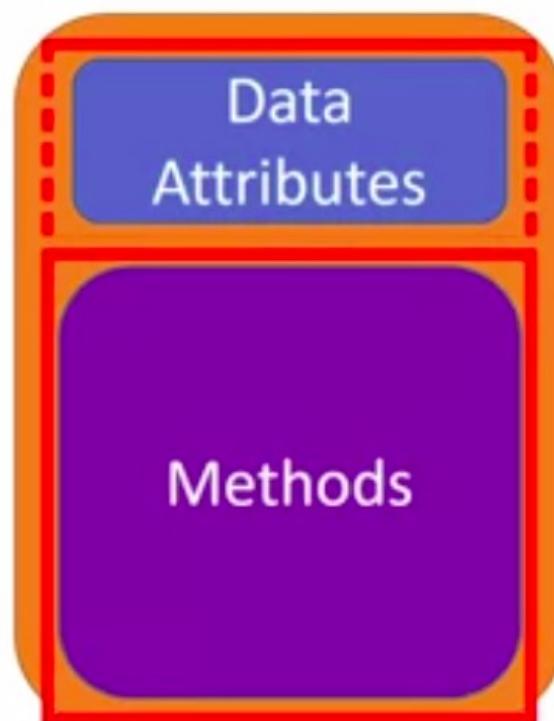
```
C1.color
```

```
"blue"
```



```
self.radius = 10;  
self.color = 'red';
```

Class



Objects or Instances of that Class



## Create a class: Circle

```
class Circle (object):  
    def __init__(self, radius , color):  
        self.radius = radius;  
        self.color = color;  
  
    def add_radius(self,r):  
        self.radius= self.radius +r
```

## Create an instance of a class: Circle

```
C1=Circle(2,'red')
```

```
C1.add_radius(8)
```

```
self.radius = 2  
self.color ='red'
```

```
def add_radius(self,8):  
    self.radius = 2 + 8  
    return (10)
```

```
self.radius = 10  
self.color ='red'
```

## Question

what is the radius attribute

after the following code block is run:

```
1 BlueCircle=Circle(10,'blue')
2 BlueCircle.add_radius(20)
```

- 10
- 20
- 30

 **Correct**  
correct

Skip

Continue

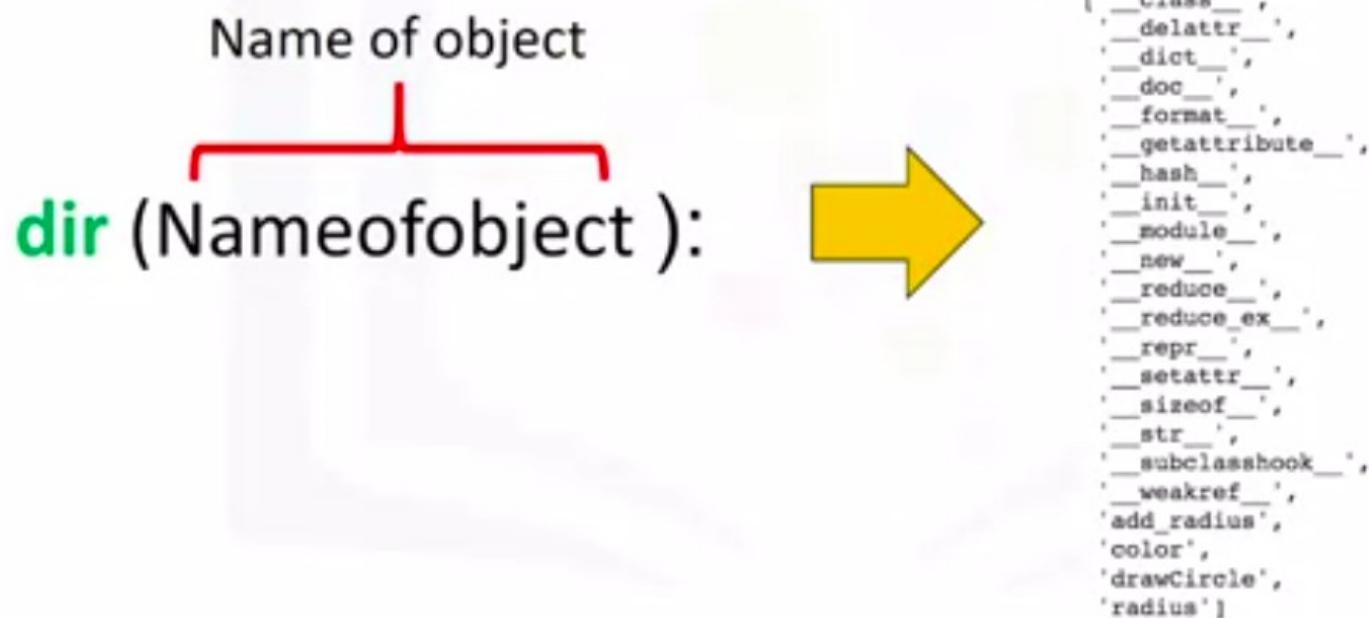
## Create a class: Circle

```
class Circle (object):  
    def __init__(self, radius=3 , color='red'): ] We can add  
        self.radius = radius;  
        self.color = color;  
  
    def add_radius(self,r):  
        self.radius= self.radius +r  
  
    def drawCircle(self):
```

## Create a class: Circle

```
class Circle (object):  
    def __init__(self, radius=3 , color='red'): ] We can add  
        self.radius = radius; default values  
        self.color = color; for parameters  
  
    def add_radius(self,r):  
        self.radius= self.radius +r  
  
    def drawCircle(self): ] New method
```

## Create a class: Circle



[Back](#)

## Practice Quiz

Practice Quiz • 15 min • 5 total points

1. What is the type of the following?

1 / 1 point

["a"]

 str list

Correct

Correct, the "a" is surrounded by brackets so it is a list.

2. What does a method do to an object?

1 / 1 point

 Changes or interacts with the object Returns a new values

Correct

correct

Back

## Practice Quiz

Practice Quiz • 15 min • 5 total points

3. We create the object:

```
Circle(3,'blue')
```

1 / 1 point

What is the color attribute set to?

 2 'blue'

Correct

Correct.

4. What is the radius attribute after the following code block is run?

1 / 1 point

```
RedCircle=Circle(10,'red')
```

```
RedCircle.radius=1
```

[Back](#)

## Practice Quiz

Practice Quiz • 15 min • 5 total points

4. What is the radius attribute after the following code block is run?

1 / 1 point

```
RedCircle=Circle(10,'red')
```

```
RedCircle.radius=1
```

- 10
- 1
- 'red'



Correct

Correct, the line of code **RedCircle.radius=1** will change the radius.

5. What is the radius attribute after the following code block is run?

1 / 1 point

```
BlueCircle=Circle(10,'blue')
```

```
BlueCircle.add_radius(20)
```

Back

## Practice Quiz

Practice Quiz • 15 min • 5 total points

Correct, the line of code **RedCircle.radius=1** will change the radius.

5. What is the radius attribute after the following code block is run?

1 / 1 point

```
BlueCircle=Circle(10,'blue')
```

```
BlueCircle.add_radius(20)
```

- 10
- 20
- 30

Correct

Correct.

Back

## Module 3 Graded Quiz

Graded Quiz • 30 min

Due Sep 18, 11:59 PM IST

1. What is the output of the following code?

1 / 1 point

```
1 x="Go"
2
3 if(x=="Go"):
4     print('Go ')
5 else:
6     print('Stop')
7
8 print('Mike')
```

 Go Mike Mike Stop Mike

Correct

Back

## Module 3 Graded Quiz

Graded Quiz • 30 min

Due Sep 18, 11:59 PM IST

Correct

2. What is the result of the following lines of code?

1 / 1 point

```
1 x=1  
2 x>5
```

- True  
 False

Correct

Correct

3. What is the output of the following few lines of code?

1 / 1 point

```
1 x=0  
2 while(x<2):  
3     print(x)
```

[Back](#) Module 3 Graded Quiz

Due Sep 18, 11:59 PM IST

Graded Quiz • 30 min

3. What is the output of the following few lines of code?

1 / 1 point

```
1 x=0
2 while(x<2):
3     print(x)
4     x=x+1
```

 0

1

 0

1

2

 0

1

3

4

Correct

Correct

[Back](#) Module 3 Graded Quiz

Due Sep 18, 11:59 PM IST

Graded Quiz • 30 min

4. What is the result of running the following lines of code ?

1 / 1 point

```
1 class Points(object):
2     def __init__(self,x,y):
3         self.x=x
4         self.y=y
5
6     def print_point(self):
7         print('x=',self.x, ' y=',self.y)
8
9 p1=Points("A","B")
10
11 p1.print_point()
```

- x=A
- y=B
- x=A y=B

Correct

correct

[← Back](#)

## Module 3 Graded Quiz

Graded Quiz • 30 min

Due Sep 18, 11:59 PM IST

5. What is the output of the following few lines of code?

1/1 point

```
1 for i,x in enumerate(['A', 'B', 'C']):  
2     print(i,2*x)
```

 0 AA

1 BB

2 CC

 0 A

1 B

2 C

 0 A

2 B

4 C

Correct

Correct

[Back](#) Module 3 Graded Quiz

Due Sep 18, 11:59 PM IST

Graded Quiz • 30 min

6. What is the result of running the following lines of code ?

1 / 1 point

```
6
7     def print_point(self):
8         print('x=' , self.x, ' y=' , self.y)
9
10    p2=Points(1,2)
11
12    p2.x=2
13
14    p2.print_point()
```

- x=2 y=2
- x=1 y=2
- x=1 y=1

**Correct**  
correct

[Back](#) Module 3 Graded Quiz

Due Sep 18, 11:59 PM IST

Graded Quiz • 30 min

7. Consider the function step, when will the function return a value of 1?

1 / 1 point

```
1 def step(x):
2     if x>0:
3         y=1
4     else:
5         y=0
6     return y
```

- if x is larger than 0
- if x is equal to or less than zero
- if x is less than zero

Correct

correct, the value of y is 1 only if x is larger than 0

8. What is the output of the following lines of code?

1 / 1 point

```
4     a=100
5     return(x+a)
6
7 print(f'{a}\n{a+100}')
```

[Back](#)

## Module 3 Graded Quiz

Graded Quiz • 30 min

Due Sep 18, 11:59 PM IST

8. What is the output of the following lines of code?

1 / 1 point

```
4     a=100
5     return(x+a)
6
7 print(do(1))
8
```

- 2
- 101
- 102

 **Correct**

Correct, the value of a=100 exists in the local scope of the function. Therefore the value of a=1 in the global scope is not used.

[Back](#) Module 3 Graded Quiz

Due Sep 18, 11:59 PM IST

Graded Quiz • 30 min

9. Write a function name **add** that takes two parameter **a** and **b**, then return the output of **a + b** (Do not use any other variable! You do not need to run it. Only write the code about how you define it.)

```
1 def add(a,b):  
2     return(a+b)
```

[Run](#)[Reset](#)

Good job!

10. Why is it best practice to have multiple except statements with each type of error labeled correctly?

1 / 1 point

- Ensure the error is caught so the program will terminate
- In order to know what type of error was thrown and the location within the program
- To skip over certain blocks of code during execution
- It is not necessary to label errors

